Wireless V2I Communication Protocols for Driver Assistance

School of Science and Computing, Waterford Institute of Technology, Waterford, Ireland

College of Science and Engineering, Central Michigan University, Mt. Pleasant, USA

Contents

Abstract

The project presents an application of utilization of multiple IoT applications. The project will be based around the idea driver safety in the real world. There will be the use of a Raspberry Pi that will be programmed to transfer from different traffic light states (red, yellow, or green). There is a circuit in which displays the traffic light signals using red, yellow and green LED's. The light systems will be pushed to an MQTT server broker. On the vehicle side, we will utilize a Raspberry Pi car. The Raspberry Pi car will be also connected to the MQTT broker. The car will constantly be searching for the traffic lights state from the broker. Once a signal is received, the car will be programmed to adjust its speed accordingly. Finally, the traffic light will have a GPS chip connected to it. That sensor will be transmitting its location to the Raspberry Pi on the car. Using the data, it has received, it will make its decision on stopping or travelling through the light.

Introduction:

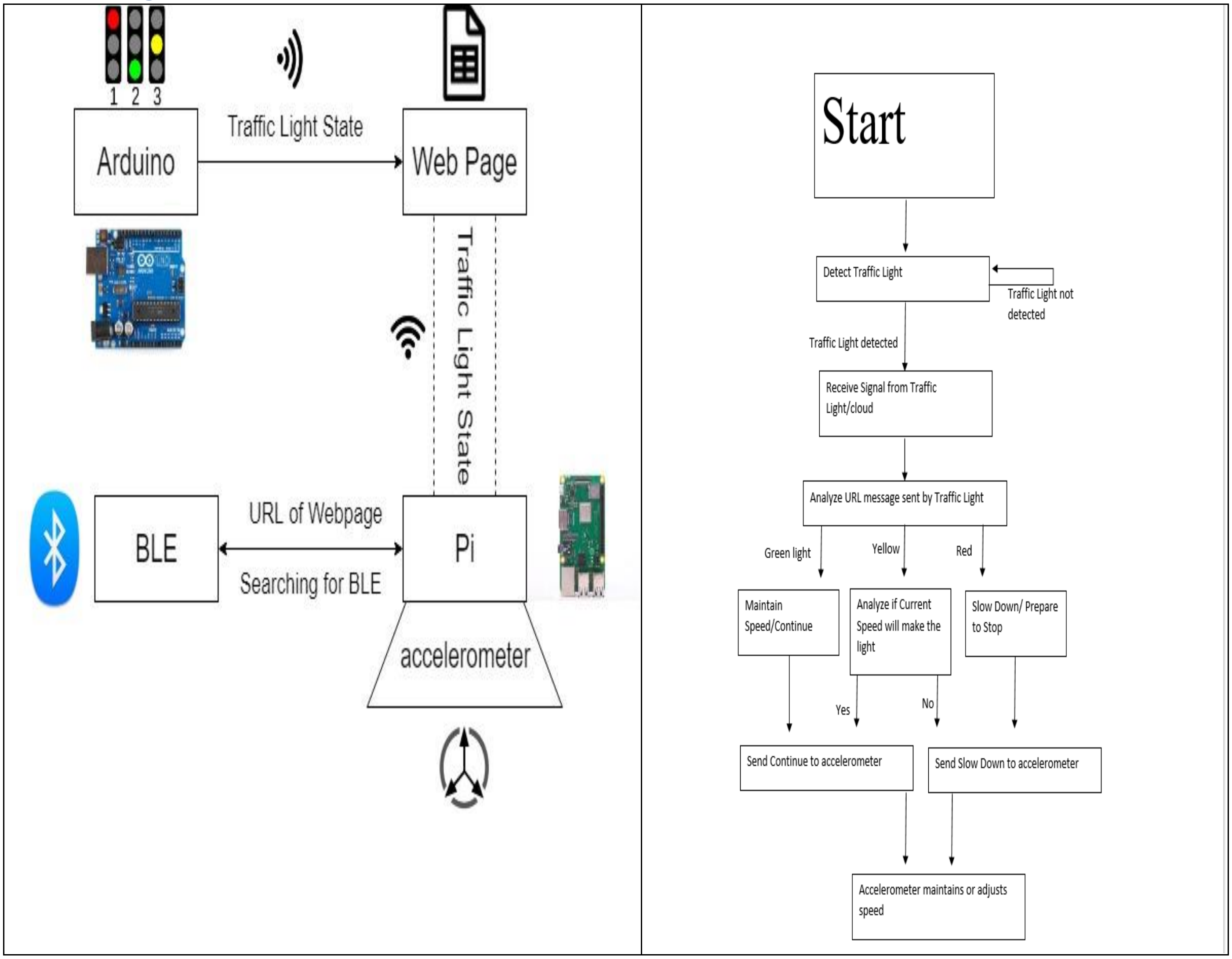
The main purpose of the project is to assist the driver when approaching traffic lights by notifying the driver in advance of the state of the of the traffic lights ahead. The proposed concept is applicable in an inner-city area, where speeds are lower which allows manipulation of the vehicles speeds possible. Also, the slower speeds will allow the GPS location to be more accurate.

For a visual display of the traffic lights we will use an Raspberry Pi circuit connected to red, yellow and green LEDs. The Raspberry Pi will also be programmed to control the states of the traffic lights system changing periodically to mimic an actual traffic light system. The traffic light states will be pushed to an online MQTT broker. This broker will be the connection between the traffic light and the vehicle which is approaching it.

The MQTT broker will search for a device to connect with. When it finds the Raspberry Pi onboard the vehicle it will send a traffic light status update to the Raspberry Pi. The Raspberry Pi, which will have access to the internet, will read the status and determine the vehicles next movements. The vehicle will then determine the GPS distance between itself from the GPS sensor located on the traffic light. Depending on the location and traffic light status, it will determine if it need to continue accelerating, slow down, or stop the vehicle completely.

An RC car could serve as a substitute for a vehicle for testing. An accelerometer could be attached to the RC car. The motors of the car can then be signaled to slow down or stop if the lights ahead are amber or red. If the lights are green the car can be notified to continue. The traffic light will be displayed using a Raspberry Pi traffic light simulator.

Block Diagram



Scope

For the successful completion of this project, there are some key areas to focus on. Connecting the Raspberry Pi to a MQTT broker service. Then using the MQTT broker as the data transfer application, it will have a traffic state ready to be broadcasted to the vehicle. The vehicle will then decide based on the state using distance from the light as a factor as well.

Once the data is on the MQTT broker, the next step is having the vehicle receive the data. The vehicle will constantly be searching for the signal from broker. All of this is happening while the vehicle is also receiving the data from the GPS sensor on the traffic light. A formula will be used to determine the vehicles next action based on the conditions.

When the data is parsed, relevant corrective action needs to be taken with respect to the state of the light. Using the speed calculated by the accelerometer, if the car is deemed to be travelling too fast, the motors will be slowed down by the Raspberry Pi. The car will be fully autonomous when coming up to the traffic light and will not need user assistance to stop the vehicle.

Rationale

The conceptualization of this project was based around the conversion of the typical car from manual to autonomous, while also trying to lower human error as a cause of car crashes. This idea could be easily manipulated to include all forms of traffic signs/signals.

With the constant amount issues of vehicular traffic, especially in highly populated areas this could be utilized to minimize traffic violations, speed up traffic, and inevitably make driving a more safe, enjoyable experience for any of the users.

Another motive for choosing this project is the crossover between the Computer Science students from WIT and the Engineering students from CMU. By constructing an RC car, we are bringing in the skills of the Engineering students to link the Raspberry Pi to the motors and accelerometer and using the Raspberry Pi board with connections to the MQTT broker. Utilizing the skills from all members of the team, we will be able to make this project a viable option for traffic safety. All members will also get to expand on their knowledge which can be beneficial as we progress our learning.

Goals

There are three main goals in conducting this project. Firstly, we would like to evaluate the usefulness of the GPS sensor. We would like to test and track its accuracy and to highlight its potential strengths and weaknesses. One example of a weakness from a previous study being "Moreover, most of the assessments were done on a predefined route or by simulations. Furthermore, none of the solutions applies information about an actual speed limit of road" [3]. The success of this goal will be determined by our ability to establish a connection and transmit the location. This can cause issues with establishing a speed of the vehicle. If the GPS unit does not track frequently it may provide us with data thats to far from the past. Ways to eliminate this research are being researched.

Another goal of conducting our project is to circumvent or limit the time-consuming effect of the handshake protocol. "Measuring handshake time between devices is an important test when it comes to implementing a communication technology in a vehicular environment characterised by a small window in time in which a vehicle can transmit and receive data to/from road infrastructure equipment. As in many applications the amount of data that is to be transmitted is not high, the communication handshake appears to be the main parameter to be considered in deciding the proper communication technology to be implemented in vehicular environment" [2]. We can evaluate the success of this goal by the ability of our Raspberry Pi to connect to the MQTT broker and interpret the data in time. This will help eliminate and test possible accuracy errors due to having poor timing. Poor timing can result in false data transmission due to the constant changing in the traffic light.

Lastly, we have the goal of successfully implementing an accelerometer onboard an RC car which will measure the acceleration of the car, using the car as a model replacing a real car in our testing process. Considering the speed measured by this sensor we could slow down or stop the motors in the RC car in accordance with the state of the traffic lights. The accelerometer will control how much power is being sent to the wheels. This will be the final step in the process of our project and the most important one. Since the car will always be searching for the data, it will also know when to begin accelerating again.

Results

On the software side of the project, the connection between the traffic light and raspberry pi has been established. The code to run the traffic lights and also run the code for the MQTT publisher and subscriber were written in the Python programming language. The decision to convert the code previously written to Python was due to our change from the original project design. We changed from an Arduino publishing the traffic light states to another Raspberry Pi. The team used the following broker ‘mqtt://iot.eclipse.org:1883’ to run the MQTT side of the project. This platform suited the project and worked properly and functionally throughout the development process.

On the hardware side of the project, the Raspberry Pi car from SunFounder was used. The group successfully used the Raspberry Pi’s onboard Wi-Fi to connect to the campus Wi-Fi. With this, the car was able to piggyback off a stronger Wi-Fi to receive and interpret signals from the broker at a faster rate. There was some minor difficulty in receiving the and interpreting the signal on the Raspberry Pi car. There was also code written in Python to operate the motors of the car. The car successfully moved forward in the green light state, slowed down in the yellow light state and stopped in the red-light state.

Evaluation

The group successfully completed the bulk of the project. Some of the challenges faced were hosting the traffic light state in MQTT broker, code conversion, analyzing MQTT, and the Pi car interpretation of the signal. The group worked well as a team communicating regularly throughout the week via Slack, where a project group was set up on the first week. Whenever problems were encountered the group met via Skype to work together on solving them and assigning weekly objectives, as well as to share ideas.

Discussion

The group did more research regarding the use of the GPS module to track the position of the car for improved accuracy. After searching through similar research papers, the team learned that for a similar measurement, an iCar III OBD Scan-tool adapter [3] was used to track the speed of the vehicle. This adapter reads the real time diagnostic codes and variables for the vehicle. For the project, we will just use an accelerometer which may prove to be a less accurate assessment of speed. In the same paper, the researchers also managed to connect the GPS module through UART, to a Raspberry Pi, using NMEA 0183. This gives the group a place to start searching, for a solution to the communication difficulties occurring between the Raspberry Pi car and signals received from the broker.

Limitations

The main limitations of work were the problems encountered with GPS, Raspberry Pi car wheels were not stable and Python 2/3 differences. The original plan relied on the knowledge of location, speed and state within the system. When trying to implement the GPS we encountered difficulties which influenced the overall system. The accuracy of the GPS has a discrepancy of about plus or minus ten feet. For the purposes of this project our small-scale example of a smart traffic light system would not be able to handle such a discrepancy. For this reason, the GPS portion of the project limited our capabilities. The wheels of the Raspberry Pi car were also unstable which meant that relying on the car moving straight was not possible, therefore eliminating another element of system accuracy.

Future Work

If we were to develop this project further into the future, we see many real-world uses for such a system. The potential benefits of the project were it fully operational are that it would lead to less traffic accidents. The use of such IoT systems would be a vital part of the current work in both IoT and AI. In the development of smart cities IoT systems such as smart traffic lights and smart cars will need to feed vast amounts of data into the neural networks of the AI systems running the smart cities for them to work. We feel that the vital data that could be received from the traffic lights and vehicles, could be used to great benefit for the purposes of computing the amount of traffic on certain roads and notifying drivers of potential disturbances ahead. This IoT smart traffic light system implemented correctly could co-operate with other IoT systems in the automotive sector to help achieve the goals of these sectors.

Reference Sources

[1] Gheorghiu R.A., Iordache V. & Cormos A.C. (2017) “Analysis of handshake time for bluetooth communications to be implemented in vehicular environments”. [Online] Available at: https://ieeexplore.ieee.org/document/8075955

[2] Jachimczyk B., Dziak D., Czapla J., Damps P., & Kulesza W. (2018) “IoT On-Board System for Driving Style Assessment”. [Online] Available at:
https://www.mdpi.com/1424-8220/18/4/1233/htm

[3] Bengler, Klaus. “Three Decades of Driver Assistance Systems: Review and Future Perspectives.” Three Decades of Driver Assistance Systems: Review and Future Perspectives - IEEE Journals & Magazine, https://ieeexplore.ieee.org/abstract/document/6936444
Carmona, Juan, et al. “Embedded System for Driver Behavior Analysis Based on GMM.” IEEE Xplore Digital Library, IEEE, 8 Aug. 2016, https://ieeexplore.ieee.org/abstract/document/7535365

[4]Gheorghiu R.A. & Minea M.
(2016), “Energy-efficient solution for vehicle prioritisation employing
ZigBee V2I communications”. [Online] Available at: https://ieeexplore.ieee.org/document/7754691

[5] Hallmark, Shauna, et al. “Safety Effectiveness of Stop-Sign Beacons : A Cross-Sectional Study.” Welcome to ROSA P, Midwest Transportation Center, 2 Feb. 2018, https://rosap.ntl.bts.gov/view/dot/35595

[6] Harrison, et al. “Method and Apparatus for Providing a Proximity Alert to the Operator of a Vehicle.” Free Patents Online, 4 Sept. 2018, www.freepatentsonline.com/10068473.html.

[7] Iordache V., Gheorghiu R.A., Minea M. & Cormos A.C. (2017), “Field testing of Bluetooth and ZigBee technologies for vehicle-to-infrastructure applications”. [Online] Available at: https://ieeexplore.ieee.org/document/8246274

[8] Lee, Jin-Shyan, et al. “A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi.” A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi - IEEE Conference Publication, 3 Nov. 2007, https://ieeexplore.ieee.org/abstract/document/4460126

[9] Pu, Cuiping, et al. “Design of Vehicle
Tracking System Based on Ibeacon.” IEEE Xplore Digital Library, IEEE, 2017,
www.revistadelafacultaddeingenieria.com/index.php/ingenieria/article/download/2115/2075